

L1: Entry 1 of 1

File: JPAB

Jun 8, 1988

PUB-NO: JP363136259A

DOCUMENT-IDENTIFIER: JP 63136259 A

TITLE: VECTOR PROCESSOR

PUBN-DATE: June 8, 1988

INVENTOR-INFORMATION:

NAME	COUNTRY
TAKAHATA, MASAMI	
AOKI, YUJI	

ASSIGNEE-INFORMATION:

NAME	COUNTRY
HITACHI LTD	
HITACHI COMPUT ENG CORP LTD	

APPL-NO: JP61281820

APPL-DATE: November 28, 1986

INT-CL (IPC): G06F 15/347; G06F 12/08; G06F 12/10

ABSTRACT:

PURPOSE: To perform conversion of addresses without deteriorating the throughput of memory access by arranging plural table reference requests into a single one and referring to the same entry of an address conversion table.

CONSTITUTION: The logical addresses a0~a3 which are calculated in parallel by address adders 5-0~5-3 are set at registers 6-0~6-3 with a synchronizing signal 19. The outputs of registers 6-0 and 6-2 are compared with the outputs of registers 6-1 and 6-3 by comparators 7-0 and 7-1. The comparison results 21-0(=0) and 21-1(=1) are delivered since a0≠a2 and a1=a3 are satisfied. In case the element space value is smaller than 1/3 block size, the address selection signals 24-0 and 24-1 are set at 0 and 1 respectively. Thus addresses a0 and a3 are supplied to address conversion tables 9-0 and 9-1 respectively and physical addresses b0 and b1~b3 are obtained. The selection signals 13-0 is set at 0 in a table reference control circuit 10 together with selection signal 13-1~13-3 set at 1 respectively. Thus the addresses b0, b1~b3 received from each table are sent to a main memory control mechanism SCU via registers 16-0~16-3.

COPYRIGHT: (C)1988, JPO&Japio

## ⑪ 公開特許公報 (A)

昭63-136259

⑤Int.Cl.  
G 06 F 15/347  
12/08  
12/10

識別記号

厅内整理番号  
A-8320-5B  
U-7927-5B  
C-7927-5B

⑩公開 昭和63年(1988)6月8日  
審査請求 未請求 発明の数 1 (全6頁)

④発明の名称 ベクトル処理装置

⑪特 願 昭61-281820

⑪出 願 昭61(1986)11月28日

⑫発明者 高畠 正美 神奈川県秦野市堀山下1番地 日立コンピュータエンジニアリング株式会社内  
⑫発明者 青木 雄二 神奈川県秦野市堀山下1番地 株式会社日立製作所神奈川工場内  
⑪出願人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地  
⑪出願人 日立コンピュータエンジニアリング株式会社 神奈川県秦野市堀山下1番地  
⑫代理人 弁理士 小川 勝男 外1名

## 明細書

## 1. 発明の名称

ベクトル処理装置

## 2. 特許請求の範囲

(1) 要素並列制御方式をとるベクトル処理装置において、複数要素の論理アドレスを並列に計算する複数のアドレス計算手段と、アドレス計算の結果得られる論理アドレスを物理アドレスに変換する前記アドレス計算手段と異なる個数の複数のアドレス変換手段と、前記複数のアドレス計算手段から送出される論理アドレス列を相互に比較し、該論理アドレス列が論理アドレス空間の局所空間に属するとき、複数の論理アドレス列から1個のアドレス変換リクエストを生成する制御手段と、前記複数のアドレス計算手段から送出される論理アドレスがおのおの異なる局所空間に属するとき、アドレス列生成ピッチをアドレス変換ピッチに一致するように減少せしめる制御手段を具備していることを特徴とするベクトル処理装置。

## 3. 発明の詳細な説明

## 〔産業上の利用分野〕

本発明はベクトル処理装置に係り、特に要素並列制御方式のベクトル処理装置におけるアドレス変換機構に関する。

## 〔従来の技術〕

大規模の科学技術計算に使用されるベクトル処理装置は大容量の主記憶と高い演算スループットとを特徴とするが、主記憶へのアクセスについてはアドレス計算結果をそのまま使用する実アドレス方式であるために多大なプログラミング環境下における主記憶の利用効率は十分ではなかった。

従来、ベクトル処理装置におけるユーザ・ジョブの動的再配賦については、基底レジスタによるアドレス修飾によってロード・モジュール内のアドレスの再計算を実行時に用いる方式があるが(リヤード・M・ラッセル: CRAY-1コンピュータ・システム、日経マグロウヒル、P. 290~292, 1982)、ページングを可能とするような動的アドレス変換機構による仮想記憶方

式には至っていない。

〔発明が解決しようとする問題点〕

ベクトル処理装置において、動的アドレス変換機構の構成を困難にしている理由は、アドレス変換を伴うオペランド参照の高スループットを確保できない点にある。ベクトル処理装置においては統続的に高い演算スループットを発揮させるためには、オペランド参照におけるアドレス計算、アドレス変換、メモリ・リクエスト発行のすべてについて十分なスループットが必要である。要素並列制御方式のベクトル処理装置では、通常、オペランド参照のスループットに応じて並列に処理される要素が定まり、上記の構成要素が多重化される。したがって、並列要素数が増大する毎、ハードウェア量が増大する。

本発明の目的は、要素並列制御方式のベクトル処理装置において、主記憶上に等間隔に配設されるベクトル・データのアクセスについて、並列要素数に依存しないハードウェア量にて最小限のスループット低下でアドレス変換を処理する動的ア

ドレス変換機構を有するベクトル処理装置を提供することにある。

〔問題点を解決するための手段〕

本発明は、先頭要素アドレスと要素間隔値によって記述されるベクトル・データの複数の要素の論理アドレスを並列に計算するアドレス加算器と、アドレス計算の結果得られる論理アドレスを主記憶上のアドレス付けに用いられる物理アドレスに変換する最長2個のアドレス変換テーブルと、並列に計算された複数の論理アドレスを比較し、比較結果に基づいて複数のテーブル参照要求を1個にまとめ、アドレス変換テーブルに入力し、複数の物理アドレスを同時に生成するテーブル参照制御回路と、アドレス変換の結果得られる複数の物理アドレスの同期を取り、主記憶制御機構に対してフェッチ、ストア等のアクセス要求を発行するメモリ・リクエスト制御回路と、ベクトル・データの要素間隔値をブロック・サイズと比較し、テーブル参照要求を圧縮できない場合を検出し、アドレス計算、メモリ・リクエスト発行のピッチ

を変化させるリクエスト・ピッチ制御回路とを設ける。

〔作用〕

複数のテーブル参照要求を1個にまとめることは可能なのは、ベクトル・データの要素間隔がブロック・サイズに比べて十分小さい場合である。通常、ベクトル・データの要素間隔は、扱うデータの型や計算アルゴリズムにもよるが、要素の語長の1倍、2倍、4倍等でとられる場合が多い。これに対してブロック・サイズは語長の512倍、1024倍等である。したがって、多くの場合、同一要素並列に計算されたアドレスは同一のブロックに属することが多い。この場合、複数のテーブル参照要求を1個にまとめ、アドレス変換テーブルの同一エントリを参照することによって複数のアドレスを同時に変換することが可能となり、テーブルの個数が並列要素数よりも少なくてもスループットの低下は回避される。

リクエスト・ピッチ制御回路はベクトル・データの要素間隔値をブロック・サイズと比較し、上

記テーブル参照要求の圧縮の可否を判定する。並列要素数がM個、アドレス変換テーブルがN個の場合 ( $M > N$ ) には、閾値はブロック・サイズの  $(N - 1) / (M - 1)$  である。要素間隔値が該閾値より小さい場合には同一要素並列におけるテーブル参照要求は高々N個に圧縮され、アドレス計算と同じピッチでアドレス変換を実行可能である。要素間隔値が閾値より大きい場合には同一要素並列におけるテーブル参照要求は  $N + 1$  個から最大M個となる。この場合にはアドレス変換をアドレス計算と同じピッチで実行するのは不可能である。したがって、アドレス計算のピッチを落し、アドレス変換のピッチに合せる。リクエスト・ピッチ制御回路は上記の2つの場合を識別し、アドレス加算器、テーブル参照制御回路、メモリ・リクエスト制御回路の動作ピッチを設定する。

テーブル参照制御回路は、アドレス計算によって同時に得られる複数の論理アドレスを選択し、アドレス変換テーブルを参照し、物理アドレスを生成する処理を制御する。複数のテーブル参照要

求を1個にまとめることができる場合には論理アドレスを比較し、アドレスの一致する同一ブロックに属する参照要求を1個としてアドレス変換テーブルに入力し、対応する物理アドレスを得る。同一要素並列に計算されたすべての要素が同一のブロックに属する場合にはアドレス変換テーブルに対応する1個のエントリを参照すればよい。同一要素並列に計算された前半の要素と後半の要素とが別々の2個のブロックに属する場合には2個のアドレス変換テーブルにて各々のエントリを参照すればよい。各要素とブロックとの対応付けは論理アドレスの一致によってなされ、対応付けに従って物理アドレスが生成される。複数のテーブル参照要求を1個にまとめることができない場合には、同一要素並列にアドレス計算された複数の要素のアドレス変換を(並列要素数) / 2 マシン・サイクルかけて逐次的に実行する。各要素のテーブル参照は定められた順序に従って実行される。すべての要素についてテーブル参照が完了すると、同期がとられ、メモリ・リクエストの発行

が可能となる。

#### 【実施例】

以下、本発明の一実施例について図面により説明する。

第1図は本発明を適用した要素並列制御方式のベクトル処理装置の一実施例で、特にアドレス計算に関係する部分の構成例を示したものである。第1図において、1はベクトル・データの先頭要素を保持するアドレス・レジスタ(VAR)、2は要素間隔値を保持するインクリメント・レジスタ(VIR)、3は要素間隔値をデコードするデコーダである。4はデコーダ3のデコード結果を受けて、アドレス計算、メモリ・リクエスト発行のピッチを変化させるリクエスト・ピッチ制御回路である。5はVAR1とVIR2の内容を入力として論理アドレスを計算するアドレス加算器で、こゝでは5-0~5-3の4値からなるとしている。6-0~6-3は各アドレス加算器5-0~5-3に対応して設けたレジスタである。7は2個の論理アドレスを比較するコンパレータで、こ

こでは7-0と7-1の2個からなるとしている。8-0と8-1はそれぞれ一つの論理アドレスを選択するセレクタである。9は論理アドレスを物理アドレスに変換するアドレス変換テーブルで、こゝでは9-0と9-1の2個からなるとしている。10はアドレス変換テーブル9の参照を制御するテーブル参照制御回路である。12-0~12-3はそれぞれ一つの物理アドレスを選択するセレクタ、11-0~11-3および16-0~16-3はレジスタである。15は主記憶制御機構に対してフェッチ、ストア等のアクセス要求を発行するメモリ・リクエスト制御回路である。

本実施例において、アドレス計算は4要素/1マシン・サイクルで処理される。アドレス変換テーブル参照は2要素/1マシン・サイクルで処理される。ベクトル・データの要素間隔値がブロック・サイズの1/3より小さい場合には、同一要素並列に計算された複数の論理アドレスの中に同一ブロックに属するものがあるので、実質的にテーブル参照を必要とするのは高々2要素である。

この場合には、アドレス計算、テーブル参照、メモリ・リクエスト発行を1マシン・サイクル・ピッチで処理可能である。そうでない場合には2マシン・サイクル・ピッチで処理しなければならない。

上記条件はブロック・サイズを4K = 4096バイトとしたとき  $3 \times (\text{要素間隔値}) < 4096$  として表わされる。すなわち、要素間隔値の固値は1366である。正確には要素の語長の複数倍でなければならないから、4バイト・データの場合 1364...8バイト・データの場合 1360である。

デコーダ3はVIR2の値をデコードし、リクエスト・ピッチ制御回路4に送る。リクエスト・ピッチ制御回路4は該デコード結果と語長によって異なる固値とから動作ピッチを定める。すなわち、1マシン・サイクル・ピッチ動作の場合はモード信号22を“0”とし、2マシン・サイクル・ピッチの場合は“1”とする。該モード信号22と同期化信号19とによってアドレス加算器5

におけるアドレス計算、メモリ・リクエスト制御回路15におけるメモリ・リクエスト発行のビッチを変える。テーブル参照制御回路10は別の同期化信号23によって制御される。

はじめ第1図、第2図によって1マシン・サイクル・ビッチのアドレス変換テーブル参照動作を説明する。第2図はアドレス変換テーブル9の参照ルールを示すものである。要素間隔値がブロック・サイズの1/3より小さい場合には、同一要素並列に計算された複数の論理アドレスは最大2個のブロックに分布する。このときブロックをクロスする要素位置によって、第2図の左欄の様に4つに場合分けされる。場合の識別は4要素の論理アドレスを2要素毎に比較することによって行う。比較の対となるのは0番と2番、1番と3番である。論理アドレスは要素番号順に単調に増加または減少するので、該組み合せによりブロックをクロスする要素位置を正確に認識することが可能となる。またアドレス変換テーブルを参照する論理アドレスとしては0番と3番を選べばよい。

であり、論理アドレスa<sub>0</sub>がレジスタ6-0からアドレス・セレクタ8-0を介しアドレス変換テーブル9-0に入力される。アドレス・セレクト信号24-1は“1”であり、論理アドレスa<sub>0</sub>がレジスタ6-3からアドレス・セレクタ8-1を介しアドレス変換テーブル9-1に入力される。この結果、論理アドレスa<sub>0</sub>が変換されて物理アドレスb<sub>0</sub>が、論理アドレスa<sub>3</sub>が変換されて物理アドレスb<sub>3</sub>がそれぞれ得られる。

テーブル参照制御回路10において、アドレス比較結果21-0, 21-1からセレクト信号13-0～13-3が生成され、アドレス変換テーブル参照結果の選択の制御に用いられる。今の場合、第2図より、セレクト信号13-0は“0”、13-1～13-3は“1”である。従って、アドレス変換テーブル9-0の参照結果b<sub>0</sub>はバス27-0からセレクタ12-0を介しレジスタ11-0にセットされる。アドレス変換テーブル9-1の参照結果b<sub>3</sub>はバス27-1からセレクタ12-1～12-3を介しレジスタ11-1

0番は異なる2個のブロックの前半の論理アドレスを代表し、3番は後半の論理アドレスを代表するからである。

今、同一要素並列に計算される論理アドレスをa<sub>0</sub>, a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub>に対応する物理アドレスをb<sub>0</sub>, b<sub>1</sub>, b<sub>2</sub>, b<sub>3</sub>とし、a<sub>0</sub>とa<sub>1</sub>との間でブロックをクロスするものとする。アドレス加算器5-0～5-3によって並列に計算された論理アドレスa<sub>0</sub>～a<sub>3</sub>は、リクエスト・ビッチ制御回路4が発生する同期信号19によってレジスタ6-0～6-1に毎サイクルセットされる。レジスタ6-0と6-2の出力はバス20-0, 20-2を介しアドレス・コンパレータ7-0に入力される。第2図よりa<sub>0</sub>≠a<sub>1</sub>であるから比較結果21-0は“0”となる。レジスタ6-1と6-3の出力はバス20-1, 20-3を介しアドレス・コンパレータ7-1に入力される。同様にa<sub>2</sub>=a<sub>3</sub>であるから比較結果21-1は“1”となる。また、要素間隔値がブロック・サイズの1/3より小さい場合、アドレス・セレクト信号24-0は“0”

～11-3にセットされる。レジスタ11-0～11-3はセット信号14-0～14-3によって毎サイクルセットされる。レジスタ11-0～11-3に得られた物理アドレスb<sub>0</sub>～b<sub>3</sub>はレジスタ16-0～16-3に転送され、メモリ・リクエスト制御回路15の制御の下に主記憶制御機構(SCU)に対して送出される。

第3図は以上の動作のタイム・チャートを示したものである。

次に第1図、第2図によって2マシン・サイクル・ビッチのアドレス変換テーブル参照動作を説明する。要素間隔値がブロック・サイズの1/3以上の場合には、同一要素並列に計算された複数の論理アドレスは3個または4個の異なるブロックに分布する。よって1個のアドレス変換テーブルにおいて2マシン・サイクルの間に2要素を変換することによって、全体として2マシン・サイクル・ビッチで4要素を処理するようとする。このため、第1図に示す様に、偶数番のアドレス加算器1個と奇数番のアドレス加算器1個とに対し

て1個のアドレス変換テーブルをくくり付けとする。2マシン・サイクルの内、前半では偶数番のアドレス加算器の出力を入力し、後半では奇数番のアドレス加算器の出力を入力する。両方のアドレス変換完了後、同期を取ってメモリ・リクエストを発行する。

今、同一要素並列に計算される論理アドレスを  $a_0 \sim a_3$ 、対応するアドレスを  $b_0 \sim b_3$  とする。アドレス加算器 5-0～5-3 の出力として得られる  $a_0 \sim a_3$  は、セット信号 19 によって同時にレジスタ 6-0～6-3 にセットされる。たゞしセットは2マシン・サイクル毎に行われる。次の2サイクルの前半でセレクト信号 24-0, 24-1 は “0” となり、 $a_0$  がレジスタ 6-0 からセレクタ 8-0 を介してアドレス変換テーブル 9-0 に入力され、 $a_1$  がレジスタ 6-2 からセレクタ 8-1 を介してアドレス変換テーブル 9-1 に入力される。次の2サイクルの後半ではセレクト信号 24-0, 24-1 は “1” となり、 $a_1$  がレジスタ 6-1 からセレクタ 8-0 を介してア

ドレス変換テーブル 9-0 に入力され、 $a_2$  がレジスタ 6-3 からセレクタ 8-1 を介してアドレス変換テーブル 9-1 に入力される。1サイクル経過後、セレクト信号 13-0, 13-2 が “0” となり、 $b_0$  がバス 27-0 からセレクタ 12-0 を介しレジスタ 11-0 にセットされ、 $b_1$  がバス 27-1 からセレクタ 12-2 を介しレジスタ 11-2 にセットされる。さらに1サイクル経過後、セレクト信号 13-0, 13-1 が “1” となり、 $b_1$  がバス 27-1 からセレクタ 12-1 を介しレジスタ 11-1 にセットされ、 $b_2$  がバス 27-3 からセレクタ 12-3 を介しレジスタ 11-3 にセットされる。レジスタ 11-0, 11-2 へのセットとレジスタ 11-1, 11-3 へのセットはセット信号 14-0～14-3 によって排他的に成される。

物理アドレスのレジスタ 11-0～11-3 へのセットは、偶数番の要素と奇数番の要素とで1マシン・サイクルの位相差を生じる。そこで、奇数番の要素のアドレス変換の完了時にセット信号

25 によって同時にレジスタ 16-0～16-3 にセットし、同期を取る様にする。同期化後、メモリ・リクエスト制御回路 15 の制御の下にメモリ・リクエストが SCU に対して発行される。

第4図は以上の動作のタイム・チャートを示したものである。

本実施例によれば、高々2個のアドレス変換テーブルによって、要素間隔値の小さなベクトル・データのアクセスにおけるアドレス変換をスループットの低下なく処理可能となる。

#### 【発明の効果】

本発明によれば、要素並列制御方式のベクトル処理装置において、並列要素数に依存しない個数のアドレス変換テーブルによって、動的アドレス変換を効率良く実行できる。特に主記憶上に等間隔に配置されるベクトル・データの要素間隔値がブロック・サイズに比較して小さい場合に、メモリ・アクセスのスループットの低下なくアドレス変換を実行できる。

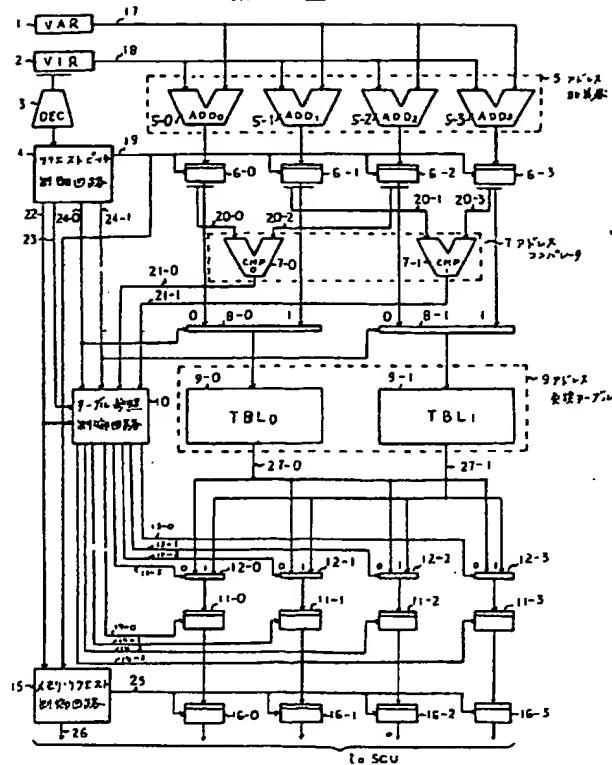
#### 4. 図面の簡単な説明

第1図は本発明による要素並列制御方式のベクトル処理装置の一実施例のプロック図、第2図はアドレス変換テーブルの参照ルールを示す図、第3図及び第4図はアドレス参照テーブルの参照動作のタイム・チャートである。

- 1 … アドレス・レジスタ、
- 2 … インクリメント・レジスタ、
- 4 … リクエスト・ピッチ制御回路、
- 5 … アドレス加算器、
- 7 … アドレス・コンバータ、
- 8 … アドレス・セレクタ、
- 9 … アドレス変換テーブル、
- 10 … テーブル参照制御回路、
- 15 … メモリ・リクエスト制御回路。

代理人弁理士 小川勝男

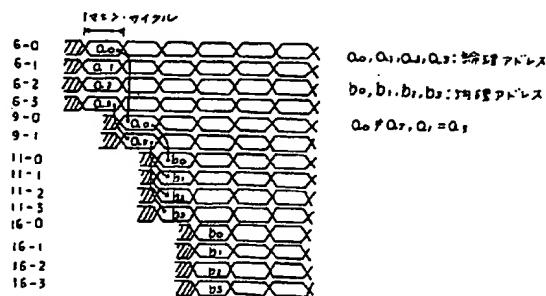
第1図



第2図

動作情報		7段表示	6段表示	5段表示	4段表示	3段表示	2段表示	1段表示
21-0	21-1	15-0	15-1	15-2	15-3	-	-	-
0	0	0	0	0	0	-	-	-
Q <sub>0</sub> = Q <sub>3</sub>	Q <sub>1</sub> = Q <sub>2</sub>	Q <sub>0</sub> = Q <sub>3</sub>	Q <sub>1</sub> = Q <sub>2</sub>	Q <sub>0</sub> = Q <sub>3</sub>	Q <sub>1</sub> = Q <sub>2</sub>	Q <sub>0</sub> = Q <sub>3</sub>	Q <sub>1</sub> = Q <sub>2</sub>	Q <sub>0</sub> = Q <sub>3</sub>
Q <sub>0</sub> = Q <sub>2</sub>	Q <sub>1</sub> = Q <sub>3</sub>	Q <sub>0</sub> = Q <sub>2</sub>	Q <sub>1</sub> = Q <sub>3</sub>	Q <sub>0</sub> = Q <sub>2</sub>	Q <sub>1</sub> = Q <sub>3</sub>	Q <sub>0</sub> = Q <sub>2</sub>	Q <sub>1</sub> = Q <sub>3</sub>	Q <sub>0</sub> = Q <sub>2</sub>
Q <sub>0</sub> = Q <sub>1</sub>	Q <sub>2</sub> = Q <sub>3</sub>	Q <sub>0</sub> = Q <sub>1</sub>	Q <sub>2</sub> = Q <sub>3</sub>	Q <sub>0</sub> = Q <sub>1</sub>	Q <sub>2</sub> = Q <sub>3</sub>	Q <sub>0</sub> = Q <sub>1</sub>	Q <sub>2</sub> = Q <sub>3</sub>	Q <sub>0</sub> = Q <sub>1</sub>
Q <sub>0</sub> ≠ Q <sub>3</sub>	Q <sub>1</sub> ≠ Q <sub>2</sub>	Q <sub>0</sub> ≠ Q <sub>3</sub>	Q <sub>1</sub> ≠ Q <sub>2</sub>	Q <sub>0</sub> ≠ Q <sub>3</sub>	Q <sub>1</sub> ≠ Q <sub>2</sub>	Q <sub>0</sub> ≠ Q <sub>3</sub>	Q <sub>1</sub> ≠ Q <sub>2</sub>	Q <sub>0</sub> ≠ Q <sub>3</sub>
Q <sub>0</sub> ≠ Q <sub>2</sub>	Q <sub>1</sub> ≠ Q <sub>3</sub>	Q <sub>0</sub> ≠ Q <sub>2</sub>	Q <sub>1</sub> ≠ Q <sub>3</sub>	Q <sub>0</sub> ≠ Q <sub>2</sub>	Q <sub>1</sub> ≠ Q <sub>3</sub>	Q <sub>0</sub> ≠ Q <sub>2</sub>	Q <sub>1</sub> ≠ Q <sub>3</sub>	Q <sub>0</sub> ≠ Q <sub>2</sub>
Q <sub>0</sub> ≠ Q <sub>1</sub>	Q <sub>2</sub> ≠ Q <sub>3</sub>	Q <sub>0</sub> ≠ Q <sub>1</sub>	Q <sub>2</sub> ≠ Q <sub>3</sub>	Q <sub>0</sub> ≠ Q <sub>1</sub>	Q <sub>2</sub> ≠ Q <sub>3</sub>	Q <sub>0</sub> ≠ Q <sub>1</sub>	Q <sub>2</sub> ≠ Q <sub>3</sub>	Q <sub>0</sub> ≠ Q <sub>1</sub>

第3図



第4図

